Simulation based Optimization

Eldad Haber

August 18, 2008

Goals

- PDE optimization problems can be very involved.
- Try to explain the essence and possible pitfalls
- Encourage you to get into this *cool!* field
- Give some simple software to demonstrate these concepts

Outline

Outline

- Introduction
- Difficulties and PDE aspects
- The optimization framework
- Solving the KKT system
- Optimization algorithms
- Examples
- Summary and future work

Matlab Code

Outline

- Introduction
- Difficulties and PDE aspects
- The optimization framework
- Solving the KKT system
- Optimization algorithms
- Examples
- Summary and future work

Matlab Code

Outline

- Introduction
- Difficulties and PDE aspects
- The optimization framework
- Solving the KKT system
- Optimization algorithms
- Examples
- Summary and future work

Matlab Code

Outline

- Introduction
- Difficulties and PDE aspects
- The optimization framework
- Solving the KKT system
- Optimization algorithms
- Examples
- Summary and future work

Matlab Code

Outline

- Introduction
- Difficulties and PDE aspects
- The optimization framework
- Solving the KKT system
- Optimization algorithms
- Examples
- Summary and future work

Matlab Code

Outline

Outline

- Introduction
- Difficulties and PDE aspects
- The optimization framework
- Solving the KKT system
- Optimization algorithms
- Examples
- Summary and future work

Matlab Code

Simulation and Optimization

The (continuous) problem:

$$\begin{aligned} \min \quad & \mathcal{J}(y,u) \\ \text{subject to} \quad & c(y,u) = 0 \end{aligned}$$

$u \in \mathcal{U}$ model - control
$y \in \mathcal{Y}$ field - state
$\mathcal{J} : [\mathcal{U} \times \mathcal{Y}] \rightarrow \mathcal{R}^1$
$c : [\mathcal{U} \times \mathcal{Y}] \rightarrow \widehat{\mathcal{Y}}$

Simulation and Optimization

The (discrete) problem:

$$\begin{array}{rl} \min & \mathcal{J}(y,u) \\ \text{subject to} & c(y,u) = 0 \end{array}$$

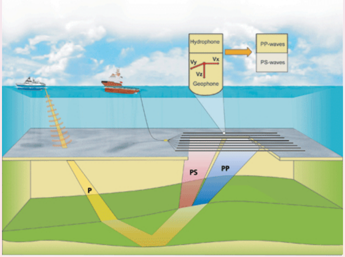
$u \in \mathcal{R}^n$ model - control
$y \in \mathcal{R}^m$ field - state
$\mathcal{J} : \mathcal{R}^{m+n} \to \mathcal{R}^1$
$c : \mathcal{R}^{m+n} \to \mathcal{R}^m$

Example I

Seismic inversion Clerbout 2000

$$\min \qquad \mathcal{J} = \frac{1}{2}\sum_{i} \|Q_j y_j - d\|^2 + \frac{\alpha}{2}\|Lu\|^2$$

$$\text{s.t.} \qquad c(y_j, u) = \Delta_h y_j + k^2 u \odot y_j = 0 \qquad j = 1, \ldots n_s$$

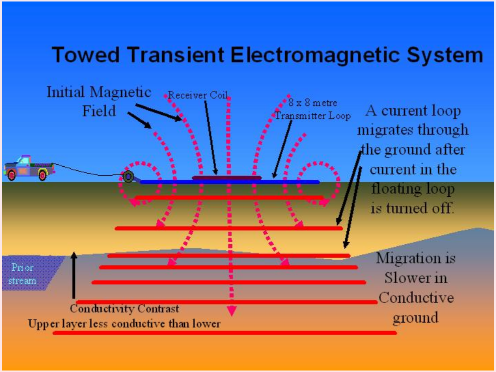


Example II

Electromagnetic inversion Newman 1996

$$\min \quad \mathcal{J} = \frac{1}{2}\sum_j \|Q_j y_j - d\|^2 + \frac{\alpha}{2}\|Lu\|^2$$

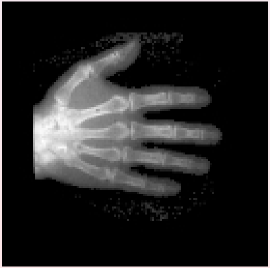
$$\text{s.t.} \quad c(y_j, u) = (\nabla \times \ \mu^{-1}\nabla\times\ )_h y_j + i\omega S(u) y_j = 0 \quad j = 1, \ldots, n_s$$



Example III

Image Processing - transprot Modersitzki 2003

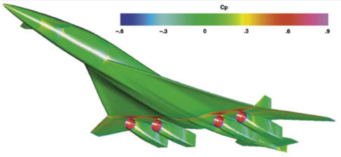
$$\begin{array}{rl} \min & \mathcal{J} = \frac{1}{2}\|y(T,x) - d(x)\|^2 + \alpha S(u) \\ \text{subject to} & y_t + u^\top \nabla y = 0 \qquad y(0,x) = y_0(x) \end{array}$$



Example IV

Shape Optimization Haslinger & Makinen 2003

$$\begin{array}{rl} \min & \mathcal{J} = g(y) \\ \text{subject to} & c(y,u) = \Delta_h y - f(u) = 0 \end{array}$$



Some Historical Prespective

Optimization with O/PDE constraint is common practice in many applications for many years

- Geophysical inversion for conductivity (Schlumberger 1912)
- **Other fields:** Flow design, VLSI, trajectory planning, chemical reaction control, ... (starting in the 30's and on)

However,

- better computer architecture $\longrightarrow$ larger simulations
- development in numerical PDE's $\longrightarrow$ complex models

New optimization algorithms are needed

Some Historical Prespective

Optimization with O/PDE constraint is common practice in many applications for many years

- Geophysical inversion for conductivity (Schlumberger 1912)
- **Other fields:** Flow design, VLSI, trajectory planning, chemical reaction control, ... (starting in the 30's and on)

However,

- better computer architecture $\longrightarrow$ larger simulations
- development in numerical PDE's $\longrightarrow$ complex models

New optimization algorithms are needed

Some Historical Prespective

Optimization with O/PDE constraint is common practice in many applications for many years

- Geophysical inversion for conductivity (Schlumberger 1912)
- **Other fields:** Flow design, VLSI, trajectory planning, chemical reaction control, ... (starting in the 30's and on)

However,

- better computer architecture $\rightarrow$ larger simulations
- development in numerical PDE's $\rightarrow$ complex models

New optimization algorithms are needed

Some Historical Prespective

Optimization with O/PDE constraint is common practice in many applications for many years

- Geophysical inversion for conductivity (Schlumberger 1912)
- **Other fields:** Flow design, VLSI, trajectory planning, chemical reaction control, ... (starting in the 30's and on)

However,

- better computer architecture $\rightarrow$ larger simulations
- development in numerical PDE's $\rightarrow$ complex models

New optimization algorithms are needed

Before we do anything

All you got to do is think Pooh Bear

Our framework: Discretize-Optimize

$$\min \ \mathcal{J}(y,u) \qquad \text{s.t} \quad c(y,u) = 0$$

Optimize-Discretize: *Can yield inconsistent gradients of the objective functionals. The approximate gradient obtained in this way is not a true gradient of anything–not of the continuous functional nor of the discrete functional.*

Discretize-Optimize *Requires to differentiate computational facilitators such as turbulence models, shock capturing devices or outflow boundary treatment.*

M. Gunzburger

Want to use the wealth of optimization algorithms

Simulation and Optimization

- Need to discretize the PDE (constraint)
- Parameters change - modeling need to be flexible
- Need to optimize - derivatives

Discretizing $c(y,u) = 0$ - difficulties

Stability with respect to parameters

$$c(y,u) = y_t - u y_{xx}$$

Explicit vs Implicit

Explicit:

$$c_h(y_h, u_h) = y_h^{n+1} - y_h^n - u_h \odot \frac{\delta t}{\delta x^2} L y_h^n = 0$$

Discretizing $c(y,u) = 0$ - difficulties

Stability with respect to parameters

- Stability requires $u_h \delta t \approx \delta x^2$
- do not know u $\rightarrow$ hard to guarantee stability.
- Code has to make sure discretization is compatible
- Possible solution: implicit methods are unconditionally stable

Discretizing $c(y,u) = 0$ - difficulties

Stability with respect to parameters

$$c(y,u) = y_t - uy_{xx}$$

Explicit vs Implicit

Implicit:

$$c_h(y_h, u_h) = y_h^{n+1} - y_h^n - u_h \odot \frac{\delta t}{\delta x^2} L y_h^{n+1} = 0$$

No free lunch, need to invert a matrix

Discretizing $c(y, u) = 0$ - difficulties

Differentiability of the discretization

$$c(y, u) = \epsilon y_{xx} + u y_x = 0$$

Common discretization, upwind

$$\frac{\epsilon}{h^2}(y_{j+1} - 2y_j + y_{j-1}) +$$
$$\frac{1}{h}\left(\max(u_j, 0)(y_j - y_{j-1}) + \min(u_j, 0)(y_{j+1} - y_j)\right) = 0$$

Discretizing $c(y, u) = 0$ - difficulties

The continuous problem is continuously differentiable w.r.t u

$$\epsilon y_{xx} + u y_x = 0$$

The discrete problem is not differentiable w.r.t u_h

$$\begin{aligned}&\frac{\epsilon}{h^2}(y_{j+1} - 2y_j + y_{j-1}) + \\ &\frac{1}{h}\left(\max(u_j, 0)(y_j - y_{j-1}) + \min(u_j, 0)(y_{j+1} - y_j)\right) = 0\end{aligned}$$

Even more difficult for flux limiters

Discretizing $c(y, u) = 0$ - difficulties

The continuous problem is continuously differentiable w.r.t u but discrete problem is not

$$\epsilon y_{xx} + u y_x = 0$$

No magic solution for this one - can pose real difficulty for the DO approach

Discretizing $c(y, u) = 0$ - difficulties

Nonlinearity of the discretization

"the mother of all elliptic problems" Dendy 1991

$$-\nabla \cdot (u \nabla y) = q$$

Finite volume discretization

$$A(u_h) y_h = \overbrace{D^\top}^{-\nabla\cdot} \underbrace{\mathrm{diag}\,(N(u_h))}_{u} \overbrace{D}^{\nabla} y_h = q_h$$

where $N(u_h) = (A_v u_h^{-1})^{-1}$ harmonic averaging

The continuous problem is bilinear but discrete problem is more nonlinear.

Discretizing $c(y,u)=0$ - difficulties

Nonlinearity of the discretization

"the mother of all elliptic problems" Dendy 1991

$$-\nabla \cdot (u\nabla y) = q$$

Finite volume discretization

$$A(u_h)y_h = \overbrace{D^\top}^{-\nabla\cdot} \underbrace{\mathrm{diag}\,(N(u_h))}_{u} \overbrace{D}^{\nabla} y_h = q_h$$

where $N(u_h) = (A_v u_h^{-1})^{-1}$ harmonic averaging

The continuous problem is bilinear but discrete problem is more nonlinear.

Discretizing $c(y,u) = 0$ - difficulties

Nonlinearity of the discretization

$$-\nabla \cdot (u \nabla y) = q$$

Finite volume discretization

$$A(u_h) y_h = \overbrace{D^\top}^{-\nabla\cdot} \underbrace{\operatorname{diag}\left(N(u_h)\right)}_{u} \overbrace{D}^{\nabla} y_h = q_h$$

Differentiate the discrete approximation rather than the continuous one

Before we solve

- PDE optimization problems are different because PDE's are different
- To make progress need to classify them. Use similar tools for similar problems
- Need good model problems to experiment with

Discretization - summary

Classify PDE's using 2 categories

- PDE's that are smooth enough such that the DO approach works well
- PDE's that require special attention in their discretization, need OD

Although we look at the PDE through the discretization these properties are intrinsic to the PDE itself

Discretization - summary

Classify PDE's using 2 categories

- Smooth PDE's such that the DO approach works well
  - Elliptic problems
  - Parabolic problems
  - Smooth hyperbolic problems
  - Some nonlinear problems
- PDE's require special attention in their discretization, need OD
  - Hyperbolic problems with nonsmooth initial data
  - Nonlinear problems with shocks
  - Other Nonlinear problems e.g, Eikonal and alike

Accuracy issues

- For many problems, constraint must be taken seriously (physics) but the optimization less so (noise, regularization)
- In many cases the control-model change little after the first reduction of the objective function

Example:

$$\begin{aligned} \min \quad & \|u-b\|^2 + \alpha TV_\epsilon(u) \\ \text{s.t} \quad & \nabla \cdot u \nabla y = q \end{aligned}$$

where

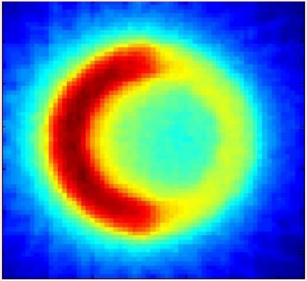
$$TV_\epsilon(t) = \begin{cases} \frac{1}{2\epsilon}t^2 + \frac{\epsilon}{2} & |t| \le \epsilon \\ |t| & |t| > \epsilon \end{cases}$$

Accuracy issues

Example:

$$
\begin{array}{rl}
\min & \|u-b\|^2 + \alpha TV_\epsilon(u) \\
\text{s.t} & \nabla \cdot u \nabla y = q
\end{array}
$$

$\epsilon = 10^0$

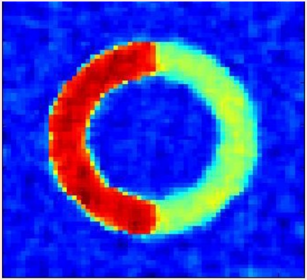


Accuracy issues

Example:

$$\begin{array}{rl} \min & \|u-b\|^2 + \alpha TV_\epsilon(u) \\ \text{s.t} & \nabla \cdot u \nabla y = q \end{array}$$

$\epsilon = 10^{-1}$

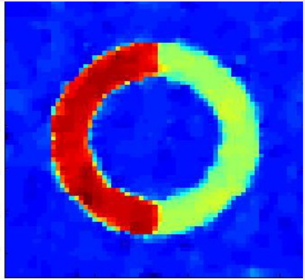


Accuracy issues

Example:

$$\begin{aligned} \min \quad & \|u - b\|^2 + \alpha TV_\epsilon(u) \\ \text{s.t} \quad & \nabla \cdot u \nabla y = q \end{aligned}$$

$\epsilon = 10^{-2}$



Optimization

Can we build it? Yes we can! Bob the builder

Solving the optimization problem

Constrained approach, solve

$$
\begin{array}{rl}
\text{min} & \mathcal{J}(y, u) \\
\text{subject to} & c(y, u) = 0
\end{array}
$$

Unconstrained approach, eliminate y to obtain

$$
\text{min} \quad \mathcal{J}(y(u), u)
$$

Constrained vs. unconstrained

Example: $c(y, u) = A(u)y - q = 0$
Constrained approach,

$$\begin{aligned} \min \quad & \mathcal{J}(y, u) \\ \text{subject to} \quad & A(u)y = q \end{aligned}$$

Unconstrained approach,

$$\min \quad \mathcal{J}(A(u)^{-1}q, u)$$

- Invertibility of $A(u)$
- Cost of evaluating the ObjFun.

Constrained vs. unconstrained

Example: $c(y,u) = A(u)y - q = 0$
Constrained approach,

$$\begin{aligned} \min \quad & \mathcal{J}(y,u) \\ \text{subject to} \quad & A(u)y = q \end{aligned}$$

Unconstrained approach,

$$\min \quad \mathcal{J}(A(u)^{-1}q, u)$$

- Invertibility of $A(u)$
- Cost of evaluating the ObjFun.

Constrained vs Unconstrained

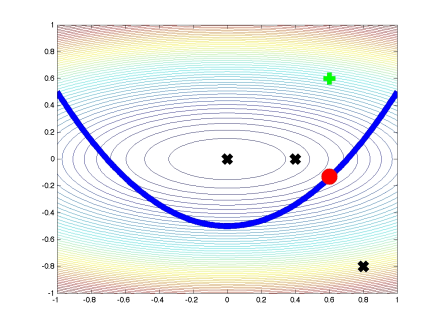
Constrained approach,

- Saddle point problem
- Algorithmically hard
- No need to solve the constraints until the end

Unconstrained approach

- Simple from an optimization standpoint
- Need to solve the constraint equation PDE
- Becomes even messier for nonlinear PDE's
- But: always feasible!!!

Constrained vs Unconstrained



Sequential Quadratic Programming

The Lagrangian

$$\mathcal{L} = \mathcal{J}(y,u) + \lambda^\top M c(y,u)$$

where

$$\lambda^\top M c(y,u) \approx \int_\Omega \lambda(x) c(y(x), u(x))\, dx$$

Differentiate to obtain the Euler Lagrange equations (Assume $M = I$)

$$\begin{aligned} \text{adjoint} \qquad & \mathcal{J}_y + c_y^\top \lambda = 0 \\ \text{state} \qquad & \mathcal{J}_u + c_u^\top \lambda = 0 \\ \text{constraint} \qquad & c(y,u) = 0 \end{aligned}$$

Computing Jacobians

- Need to compute c_y, c_u
- In many cases c_y available (used for the forward)
- Need to compute c_u, calculus with matrices helps
- In some cases c_y not used for the forward

Jacobians, example I:Hydrology, electromagnetics

$$c(y,u) = A(u)y - q = D^\top \mathrm{diag}((A_v u^{-1})^{-1}) D y - q$$

Then

$$c_y = A(u)$$
$$c_u = \frac{\partial}{\partial u}\left[D^\top \mathrm{diag}((A_v u^{-1})^{-1}) D y\right]$$

Note that

$$D^\top \mathrm{diag}((A_v u^{-1})^{-1}) D y = D^\top \mathrm{diag}(Dy)\ (A_v u^{-1})^{-1}$$

therefore

$$c_u = D^\top \mathrm{diag}(Dy)\ \mathrm{diag}((A_v u^{-1})^{-2}) A_v \mathrm{diag}(u^{-2})$$

Jacobians, example II : CFD

NS equations

$$\Delta_h y + M(y)y + \nabla_h p = u$$
$$\nabla_h \cdot y_k = 0$$

Where $M(y) \approx \nabla y$

Typical solution through fixed point iteration [Elman, Silvester, Wathen]

$$\Delta_h y_k + M(y_{k-1})y_k + \nabla_h p = u$$
$$\nabla_h \cdot y_k = 0$$

Thus to compute $c(y)$ need extra calculation

Jacobians, example II : CFD

In general

$$c(y,u) = 0$$

Use some iteration to solve (not Newton's method)
From an optimization theory we need the Jacobians c_y, c_p of the constraint otherwise cannot guarantee convergence

Open Question: Can we get away with less?

Two alternative viewpoints

$$\begin{aligned} \text{adjoint} \qquad & \mathcal{J}_y + c_y^\top \lambda = 0 \\ \text{state} \qquad & \mathcal{J}_u + c_u^\top \lambda = 0 \\ \text{constraint} \qquad & c(y,u) = 0 \end{aligned}$$

A system of nonlinear PDE's
use PDE techniques
(MG, FAS, ...)

Necessary conditions
use optimization framework
(reduce Hessian ...)

MG(linear)
MGOPT [Luis & Nash]

Two alternative viewpoints

A system of nonlinear PDE's	Necessary conditions
use PDE techniques	use optimization framework
(MG, FAS, ...)	(reduce Hessian ...)

Our approach:
Use PDE techniques as solvers
Use optimization methods for a guide

Two alternative viewpoints

A system of nonlinear PDE's
use PDE techniques
(MG, FAS, ...)

Necessary conditions
use optimization framework
(reduce Hessian ...)

Our approach:
Use PDE techniques as solvers
Use optimization methods for a guide

Solving the Euler Lagrange equations

$$\begin{aligned} \text{adjoint} \qquad & \mathcal{J}_y + c_y^\top \lambda = 0 \\ \text{state} \qquad & \mathcal{J}_u + c_u^\top \lambda = 0 \\ \text{constraint} \qquad & c(y,u) = 0 \end{aligned}$$

Approximate the Hessian and solve at each iteration the KKT system

$$\begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} & c_y^\top \\ \mathcal{L}_{yu}^\top & \mathcal{L}_{uu} & c_u^\top \\ c_y & c_u & \mathbf{O} \end{pmatrix} \begin{pmatrix} s_y \\ s_u \\ s_\lambda \end{pmatrix} = \text{rhs}$$

Solving the Euler Lagrange equations

In many applications approximate the Hessian by

$$\begin{pmatrix} \mathcal{J}_{yy} & \mathbf{O} & c_y^\top \\ \mathbf{O} & \mathcal{J}_{uu} & c_u^\top \\ c_y & c_u & \mathbf{O} \end{pmatrix} \begin{pmatrix} s_y \\ s_u \\ s_\lambda \end{pmatrix} = \mathrm{rhs}$$

Gauss-Newton SQP [Bock 89]

If $\mathcal{J}_{yy}$ and $\mathcal{J}_{uu}$ are positive semidefinite then the reduced Hessian is likely to be SPD.